

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені Тараса Шевченка
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра програмних систем і технологій

Дисципліна
«Ймовірнісні основи програмної інженерії»

Лабораторна робота № 2

Виконав:	Шекера Олександр Валерійович	Перевірила:	Вечерковська Анастасія Сергіївна
Група	ІПЗ-22	Дата перевірки	
Форма навчання	денна	Оцінка	
Спеціальність	121		
2022			

Тема: Лінійне перетворення та Графічне зображення даних

Мета: навчитись використовувати на практиці набуті знання про лінійні перетворення та графічне зображення даних.

Завдання:

1. Знайдіть Q_1 , Q_3 , P_{90}
2. Знайдіть середнє та стандартне відхилення цих оцінок.
3. Через незадоволення низькими оцінками викладач вирішив використати шкалу форми $y = ax + b$, щоб відредагувати оцінки. Він хотів, щоб середнє значення масштабних оцінок становило 95, а оцінка 100, щоб залишалася рівною 100.
4. Показати дані за допомогою діаграми "стовбур – листя".
5. Відобразити дані за допомогою коробкового графіка.
6. Зробити висновок.

Математична модель:

Формула для Q_1 :

$$pos(Q_1) = (n + 1) \frac{25}{100}$$

Формула для Q_3 :

$$pos(Q_3) = (n + 1) \frac{75}{100}$$

Формула розрахунку порядкового номера n для заданого процентиля P :

$$n = \left\lceil \frac{P}{100} N \right\rceil, 0 < P \leq 100$$

Формула пошуку середнього квадратичного відхилення:

$$s_x = \sqrt{s_x^2(x)}$$

Формула пошуку стандартного відхилення:

$$S = \sqrt{\frac{\sum_{i=1}^n (x_i - x_{cp})^2}{n - 1}}$$

Для зміни середнього значення за лінійною трансформацією використовувалась формула:

$$y = ax + b,$$

$$\text{Де : } 100 = 100a + b$$

$$\text{New Average} = \text{Mean} + b$$

Для побудови коробкового графіку використовувалась бібліотека matplotlib мови програмування python.

Псевдокод алгоритму

Пошук картелю:

START:

INIT QuartilePos = 0

IF pos equals 1

QuartilePos = (List.Size + 1) * (25/100)

RETURN List[QuartilePos]

ENDIF

ELSE IF pos equals 3

QuartilePos = (List.Size + 1) * (75/100)

RETURN List[QuartilePos]

ENDIF

END

Пошук порядкового номера n для заданного процентиля :

START

N = ((Percent * List.Size) / 100)

RETURN List[N]

END

Пошук дисперсії та середнього відхилення:

START

INIT Variation = 0

```

INIT Avarage = Avarage()
INIT Tmp[]
FOR from j = 0 to List.Size
    Tmp[j] = List[j]
ENDFOR
FOR from i = 0 to Tmp.Size
    Tmp[i] = (Tmp[i] - Avarage) ^ 2
ENDFOR
FOR from i = 0 to I Tmp.Size
    Variation += Tmp[i]
ENDFOR
Variation = Variation / Tmp.Size
INIT Difference = Sqrt(Variation)
RETURN Difference
END

```

Пошук стандартного відхилення:

```

START
    INIT Variation = 0
    INIT Avarage = Avarage()
    INIT Tmp[]
    FOR from j = 0 to List.Size
        Tmp[j] = List[j]
    ENDFOR
    FOR from i = 0 to Tmp.Size
        Tmp[i] = (Tmp[i] - Avarage) ^ 2
    ENDFOR
    FOR from i = 0 to I Tmp.Size
        Variation += Tmp[i]
    ENDFOR
    Variation = Variation / (Tmp.Size - 1)
    INIT Difference = Sqrt(Variation)
    RETURN Difference
END

```

Задача про студентів:

START

```
Avarage = Avarage()
INIT Num = 95
A = (100 - Num) / (100 - Avarage)
B = Num - Avarage * A
RETURN A, B
```

END

Код алгоритму:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.IO;

namespace Program
{
    class FileWork
    {
        private string fileName;
        private string path =
@"C:\Users\oleks\Desktop\кнх\reoprep\lab2\task_02_data\input_100.txt";
        private List<int> list = new List<int>();
        public FileWork()
        {
        }
        public FileWork(string fileName)
        {
            this.fileName = fileName;
        }
        public void Read()
        {
            StreamReader streamReader = File.OpenText(path);
            string input;
            while ((input = streamReader.ReadLine()) != null)
            {
                list.Add(Convert.ToInt32(input));
            }
            streamReader.Close();
        }
        public void Write()
        {
            var stream = new FileStream(path + "test1.txt", FileMode.Create,
FileAccess.Write);
            var writer = new StreamWriter(stream, Encoding.GetEncoding(1251));
            list.Sort();
            foreach (int num in list)
            {
                Console.Write(num + " ");
            }
            writer.WriteLine("\tRank\t\tX");
            for (int i = 0; i < list.Count; i++)
            {
                writer.WriteLine($"{i + 1}\t\t{list[i]}");
            }
            writer.WriteLine($"nQ1: {Quartile(1)}\tQ3: {Quartile(3)}\tPercentile(90):
{Percentile(90)}");
        }
    }
}
```

```

        writer.WriteLine($"\\nMean Deviation: {MeanDeviation()}\\tStandart Deviation:
{StandartDeviation()}");
        double[, ] vs = Task3(95);
        writer.WriteLine($"\\nAvarage: {Avarage()}\\ty = {vs[0,0]}x + {vs[1,1]}");
        writer.WriteLine($"Stem\\t|\\tLeaf");
        for(int i = 0; i < list.Count; i++)
        {
            if (i != 0)
            {
                if (list[i] / 10 == list[i - 1] / 10)
                {
                    if (list[i] > 99)
                    {
                        writer.Write("$", {list[i] % 100});
                    }
                    else
                    {
                        writer.Write("$", {list[i] % 10});
                    }
                }
                else
                {
                    if (list[i] > 99)
                    {
                        writer.Write($"\\n{list[i] / 10}\\t|\\t{list[i] % 100}");
                    }
                    else
                    {
                        writer.Write($"\\n{list[i] / 10}\\t|\\t{list[i] % 10}");
                    }
                }
            }
            else
            {
                if (list[i] > 99)
                {
                    writer.Write($" {list[i] / 10}\\t|\\t{list[i] % 100}");
                }
                else
                {
                    writer.Write($" {list[i] / 10}\\t|\\t{list[i] % 10}");
                }
            }
        }
        writer.Close();
    }

    private int Quartile(int pos)
    {
        double quartile_pos = 0;
        if (pos == 1)
        {
            quartile_pos = (list.Count + 1) * (25.0 / 100.0);
            return list[Convert.ToInt32(quartile_pos)];
        }
        else if (pos == 3)
        {
            quartile_pos = (list.Count + 1) * (75.0 / 100.0);
            return list[Convert.ToInt32(quartile_pos)];
        }
        else
        {
            throw new IndexOutOfRangeException();
        }
    }

    private int Percentile(int percent)
    {
        int n = ((percent * list.Count) / 100);
    }

```

```

        return list[n];
    }
    public double Avarage()
    {
        double sum = 0;
        foreach (int elem in list)
        {
            sum += elem;
        }

        return sum / list.Count;
    }
    private double MeanDeviation()
    {
        double variation = 0;
        double avarage = Avarage();
        double[] tmp = new double[list.Count];
        for (int j = 0; j < list.Count; j++)
        {
            tmp[j] = list[j];
        }
        for (int i = 0; i < tmp.Length; i++)
        {
            tmp[i] = Math.Pow((double)(tmp[i] - avarage), 2);
            variation += tmp[i];
        }
        variation = variation / tmp.Length;
        double difference = Math.Sqrt(variation);
        return difference;
    }
    private double StandartDeviation()
    {
        double variation = 0;
        double avarage = Avarage();
        double[] tmp = new double[list.Count];
        for (int j = 0; j < list.Count; j++)
        {
            tmp[j] = list[j];
        }
        for (int i = 0; i < tmp.Length; i++)
        {
            tmp[i] = Math.Pow((double)(tmp[i] - avarage), 2);
            variation += tmp[i];
        }
        variation = variation / (tmp.Length - 1);
        double difference = Math.Sqrt(variation);
        return difference;
    }
    private double[,] Task3(int num)
    {
        double[,] result = new double[2,2];
        double avarage = Avarage();
        double a = (100 - num) / (100 - avarage);
        double b = num - avarage * a;
        result[0, 0] = a;
        result[1, 1] = b;
        return result;
    }
}

```

Код побудови графіка:

```

import math
import matplotlib.pyplot as plt
import numpy as np

f = open('C:/Users/oleks/Desktop/кнху/теорвер/лаб1/task_01_data/input_100.txt', 'r')

```

```

values = []
for i in f:
    values.append(int(i))
values.sort()

plt.boxplot(values)
plt.savefig('boxplot')
plt.show()

```

Випробування алгоритму:

Для перевірки запусимо програму на файлі, з 11 значеннями:

Rank	X
1	10
2	40
3	62
4	65
5	66
6	70
7	70
8	84
9	90
10	95
11	100

Q1: 65 Q3: 95 Percentile(90): 95

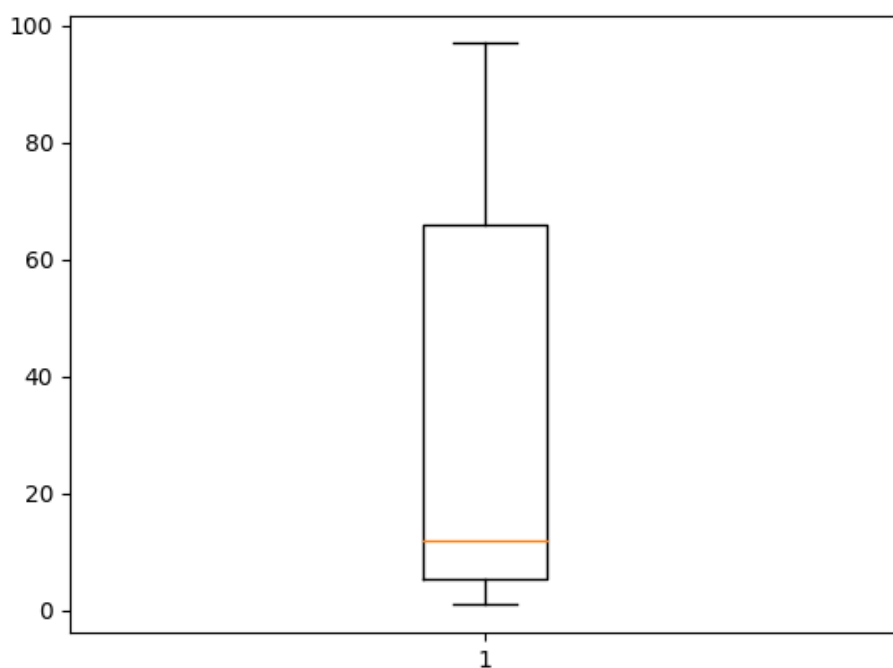
Mean Deviation: 24,6734038896974

Standart Deviation: 25,8776843139904

Avarage: 68,3636363636364

$y = 0,158045977011494x + 84,1954022988506$

Stem	Leaf
1	0
4	0
6	2, 5, 6
7	0, 0
8	4
9	0, 5
10	0



Результат роботи програми при вхідних даних розміром 100:

Q1: 38 Q3: 79 Percentile(90): 92

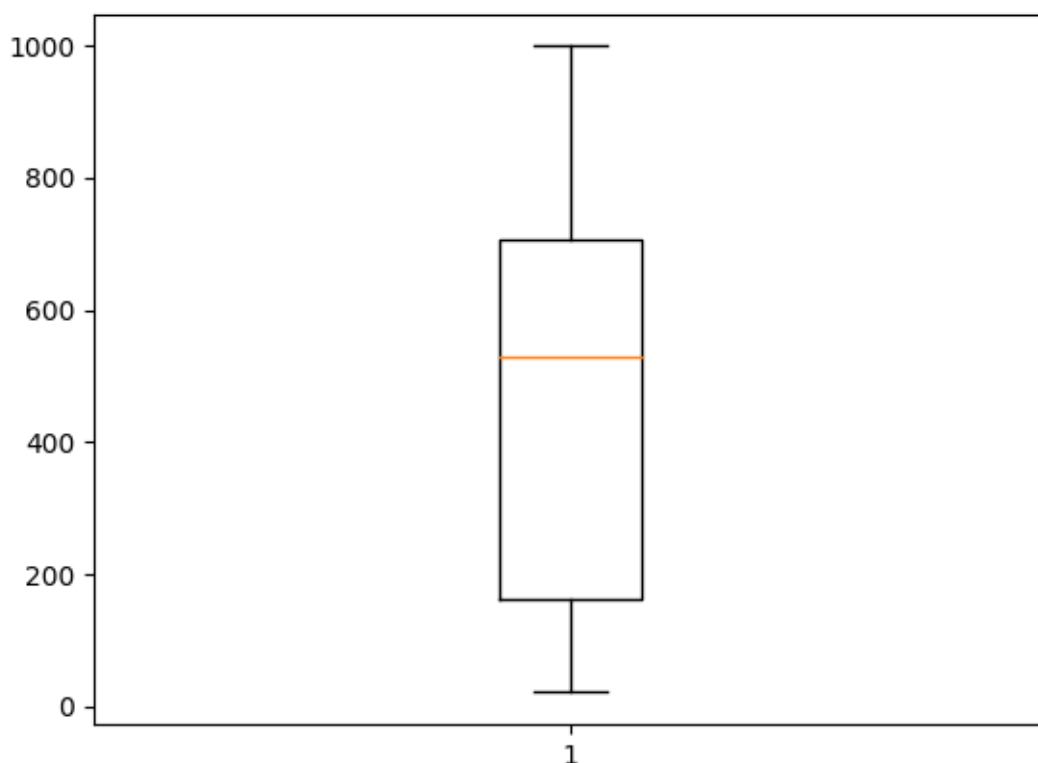
Mean Deviation: 25,3728050741276

Standart Deviation: 25,4993535153927

Avarage: 58,3762376237624

$y = 0,120123691722169x + 87,9876308277831$

Stem	Leaf
1	0, 2, 4, 4, 5, 6, 6, 9, 9
2	2, 2, 2, 2, 2, 5, 5, 5, 6, 8
3	1, 5, 5, 5, 6, 6, 8, 8
4	1, 2, 4, 5, 6, 6, 7, 9
5	0, 1, 1, 1, 2, 3, 4, 5, 6, 6, 7, 8, 8
6	0, 0, 1, 1, 1, 1, 3, 4, 5, 6, 7, 8, 8, 8, 9
7	0, 0, 1, 3, 3, 6, 6, 7, 7, 7, 8, 8, 9, 9
8	0, 1, 1, 2, 2, 2, 3, 3, 5, 7, 7, 8
9	1, 2, 2, 4, 7, 7, 8, 9, 9
10	0, 0, 0



Висновок:

Виконано завдання другої лабораторної роботи. Опановані навички використання на практиці набутих знань про лінійні перетворення та графічне зображення даних. Розроблена програма знаходить перший та третій кuartиль, 90-ий процентиль, змінює розподіл з використанням формули $y = ax + b$, відображає дані за допомогою діаграми «стовбур-листя» та коробкового графіку.